CS425 Fall 2025 – Homework 2

(a.k.a. "Once upon a time in Distributed Hollywood")

Out: Sep 19, 2025. Due: Oct 5, 2025 (2 pm US Central) (DUE on a SUNDAY!)

Topics: P2P Systems, Key-value Stores, Time and Ordering (Lectures 7-12)

Instructions:

1. Please double-check that the top of this page mentions the correct semester you are taking the course in (otherwise, you are looking at an old version and we will not accept your submitted solutions).
2. **Attempt any 8 out of the 10 problems** in this homework (regardless of how many credits you're taking the course for). If you attempt more, we will grade only the first 8 solutions that appear in your homework (and ignore the rest). Choose wisely!
3. Please hand in **solutions that are typed** (you may use your favorite word processor. We will not accept handwritten solutions. Figures (e.g., timeline questions) and equations (if any) may be drawn by hand (and scanned).
4. **All students (On-campus and Online/Coursera)** – Please submit PDF only! Please submit on Gradescope. [https://www.gradescope.com/]
5. Please **start each problem on a fresh page**, and **type your name at the top of each page**. **And on Gradescope please tag each page with the problem number!**
6. Homeworks will be **due at time and date noted above. No extensions. For DRES students only:** once the solutions are posted (typically a few hours after the HW is due), subsequent submissions will get a zero**. All non-DRES students must submit by the deadline time+date.**
7. Each problem has the same grade value as the others (10 points each).
8. Unless otherwise specified in the question, the only resources you can avail of in your HWs are the provided course materials (slides, textbooks, etc.), and communication with instructor/TA via discussion forum and e-mail.
9. You can discuss lecture concepts and the questions on Piazza and with your friends, but you cannot discuss solutions or ideas on Piazza.

Prologue: You have just been made the technical head in a production company that is producing a new Hollywood movie. The movie is sure to be a blockbuster, with a lot of well-known actors and actresses hired to star in it. Amazingly many of them know distributed systems! You run into them every day on the set. Here is what ensues.

All characters and their actions used in this homework are meant to make the homework fun! Any resemblance of their actions or opinions to real events, or places, is purely coincidental. Any stories involving real actors or actresses are fictional.

Problems:

1. Walt Disney and Pokemon want to build a virtual theme park where young customers will all wear VR goggles and be connected via a Gnutella P2P system. At one point of time, the Gnutella topology looks like a perfect balanced binary tree with $N=2^m-1$ processes. (m large). All leaves are at the same level.
   a. What is the minimum TTL required so that a query is received by all, no matter who its sender is?
   b. How many processes (apart from the sender) receive the query if a child of the root is the querying node, and the TTL=m-2?
   c. What is the TTL required by a child of the root for everyone to receive its query?

2. You want to make a movie on US Presidential elections. You are putting together the movie's cast. To facilitate the search, you decide to use a Chord-like peer-to-peer system to connect the mobile phones of aspiring artists with each other. The Chord ring has m = 12. Nodes with the following peer ids (or node ids) join the system: 1996, 2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032 (those are all years that the Presidential Elections are/will be held). Then, answer the following questions:
   a. Show or list all finger table entries for node 2020.
   b. When all finger tables and successors have converged, show the path taken by a search (or query) message originating from node 2016 intended for the key 1999.
   c. Node 2016 fails. List all the nodes whose finger tables need to be updated.

3. To connect the movie stars with its fans, you design a different peer-to-peer network that results in better lookup latencies than Chord. So, you design a peer-to-peer system based on Kelips, called Cinelips. Cinelips has a large number of nodes N (N large), but only 10 (fixed) affinity groups. Otherwise, internal protocols of Cinelips are similar to Kelips (gossip, affinity groups, contacts—all of it, i.e., the works!). Assuming no failures:
   a. What is the worst-case lookup cost in Cinelips in O()?

b. What is the memory usage in Cinelips? (O() notation)
c. Give one advantage of Cinelips over Kelips, and one advantage of Kelips over Cinelips.
d. Somone argues that if you really want to reduce lookup latency it is better to use Cassandra's design. Are they right?

4. One of the producers, Leo Bloom, likes Bloom filters. In order to make more money, he decides to make the film a flop. His mind at ease, he uses his spare time to create a Bloom filter uses m=32 bits, and 4 hash functions h1, h2, h3, and h4, where $hi(x) = (x + x^i) \bmod m$. His program then starts inserting continuous integers starting from 2024, 2025, 2026, 2027, …. and so on. Before inserting each integer, his program checks if it is already in the Bloom filter (i.e., is a false positive)—if it is not, then the integer is inserted; if it is a false positive, the program terminates. What integer does the program terminate on? (Give the integer that is the false positive, not the last-inserted integer).

5. A big-budget movie is being directed by N directors. The N directors never seem to agree on anything about the movie. To solve this, the producer decides to implement a quorum approach. In a system of N processes (N large enough), THREE quorum sets Q1, Q2, and Q3 are selected in an arbitrary manner, all of the same size M.
   a. What should this minimum value of M be so that at least **r** processes belong to ALL three of Q1, Q2, and Q3 (that is, the intersection of all three sets is **r**)? r is a constant, much smaller than N.
   b. Can you repeat the problem if there are k quorum sets Q1, Q2, Q3, …, Qk, where k is a constant, much smaller than N.

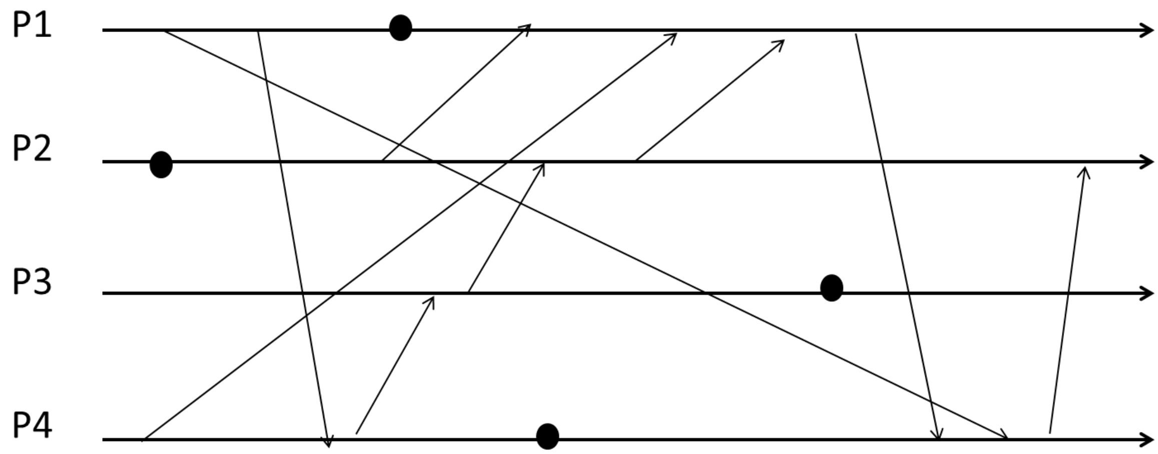
   (Hint: start with 2 sets intersecting at one process, i.e., k=2 and r=1, and quantitatively reason/think about why it has to be N/2+1).

6. The director, C. Nolan, likes to deal with time travel, which means he asks a lot of "What if?" questions. He asks you several questions about Cassandra's local storage design based on Memtables and SSTables. For each of these, give one advantage and one disadvantage:
   a. What if Cassandra didn't use tombstones for deletes and instead deleted records directly from Memtables and SSTables (but the rest of the system works the same way)?

b. What if a Cassandra replica prevents duplicate inserts, i.e., it throws an error upon put to a key that already exists in the key-value store (but the rest of the system works the same way)?
c. What if there were no Bloom filters (but the rest of the system works the same way)?
d. What if there were no compactions (but the rest of the system works the same way)?

7. Katherine Johnson was a Black computer mathematician and scientist who worked at NASA in building/writing the first NASA computer programs to do mathematical calculations for the geometrical trajectories of the Apollo moon spacecraft, being one of the first to write computer programs involving space and time. The astrophysicist consultant for your current film, a Dr. N. Tyson, is also a big fan of space and time. He is intrigued by the NTP algorithm. He poses the following question – suppose you are dealing with a network where latencies are *symmetrical* (e.g., an enhanced and controlled ATM network), i.e., latencies are identical in reverse directions, i.e., A → B message latency and B → A message latency (for same message) are known to be identical (except of course you don't know their absolute values), for every pair of processes A,B. From the equations, for NTP, can you derive the new error bound? Don't just show the last portion, please work through the entire derivation just like in the slides, from the beginning to the end.

8. Spiderman and his doppelgangers in the metaverse are trying to figure out the communication among the different metaverses. They have the following timeline of messages exchanged, where each dot represents an instruction. Can you mark *Lamport timestamps* on each event? It is ok to print out this and hand-draw/write the timestamps (then scan or photograph your solution, and insert it into your solution doc).



9. Whoops, the simultaneous and sudden arrival of all the concurrent villains (Doc Ock, Green Goblin, Electro, Sandman, and the others) means Lamport timestamps are unusable for distinguishing concurrent events from causal events. Can you mark *vector timestamps* for the same timeline as in the previous question? It is ok to print out this and hand-draw/write the timestamps (then scan or photograph your solution, and insert it into your solution doc).

10. The movie is a hit! The breakout star of the movie, Kristen, was so happy with your work that she has asked the production company to give you one last puzzle to solve before you can be paid the millions of $ you are owed. The puzzle concerns timestamps.
    a. Consider a modified version of the *Lamport timestamp* marking algorithm. Instead of incrementing timestamps by +1 each time (as in the original algorithm), the new algorithm increments it by a *random* positive integer (randomly selected for each event). Would these new timestamps still preserve (obey) causality? Describe why or why not.
    b. Consider a modified version of the *Vector timestamp* marking algorithm procedure. On an instruction or send event at process i, instead of incrementing only the i^{th} element (as in the original algorithm), a process increments all the elements in the vector. Also, on receiving a message at process i, a process instead assigns the timestamp for the event as follows:

    $$V_i[j] = \max(V_{message}[j], V_i[j]) \mathbf{+ 1} \text{ for all } j=1..N.$$

    Would these new timestamps still preserve (obey) causality? Describe why or why not.

c. Would the assigned timestamps based on the algorithm in (b) still preserve all the properties of vector timestamps? Describe why or why not.

====== END OF HOMEWORK 2 =====